The Boston Housing Dataset

Description

This dataset contains information collected by the U.S Census Service concerning housing in the area of Boston Mass. It was obtained from the StatLib archive (http://lib.stat.cmu.edu/datasets/boston), and has been used extensively throughout the literature to benchmark algorithms. However, these comparisons were primarily done outside of **Delve** and are thus somewhat suspect. The dataset is small in size with only 506 cases.

```
Variables in order:
CRIM     per capita crime rate by town
ZN       proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS    proportion of non-retail business acres per town
CHAS     Charles River dummy variable (=1 if tract bounds river; 0 otherwise)
NOX      nitric oxides concentration (parts per 10 million)
RM       average number of rooms per dwelling
AGE      proportion of owner-occupied units built prior to 1940
DIS      weighted distances to five Boston employment centres
RAD      index of accessibility to radial highways
TAX      full-value property-tax rate per $10,000
PTRATIO  pupil-teacher ratio by town
B        1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
LSTAT    % lower status of the population
MEDV     Median value of owner-occupied homes in $1000's
```

Objective

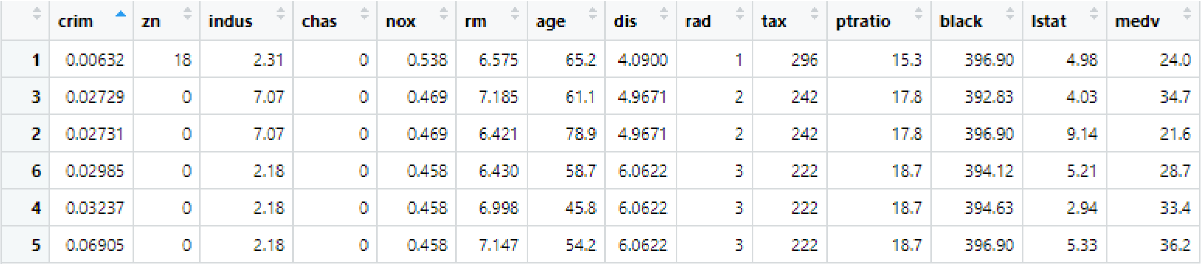
The prime objective of this project is to construct a working model which has the capability of predicting the value of houses, we will need to separate the dataset into features and the target variable. The features, ‘RM’, ‘LSTAT’, and ‘PTRATIO’, give us quantitative information about each data point. The target variable, *‘MEDV’*, will be the variable we seek to predict. These are stored in features and prices, respectively.

Overview

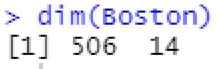
This report seeks to examine the influence of several neighbourhood attributes on the prices of housing, in an attempt to discover the most suitable explanatory variables. The specific neighbourhood attributes to be considered are proximity to the Charles River, distance to the main employment centres, pupil-teacher ratio in schools, and levels of crime. Whereas the original study focused on air pollution using nitrogen oxide concentrations as an explanatory variable, this report examines whether or not there are other, better explanatory variables for the median value of houses in Boston. The R programming language will be used to conduct this analysis.

Load Dataset

```
library(MASS)
data(Boston)
View(head(Boston))
dim(Boston)
```

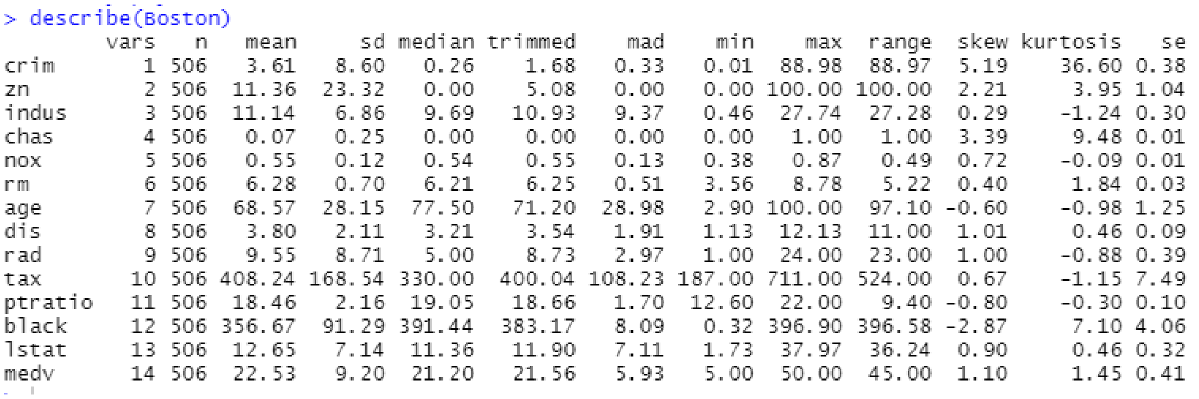


	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
1	0.00632	18	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
3	0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
2	0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
6	0.02985	0	2.18	0	0.458	6.430	58.7	6.0622	3	222	18.7	394.12	5.21	28.7
4	0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
5	0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2



```
> dim(Boston)
[1] 506  14
```

```
library(psych)
describe(Boston)
```



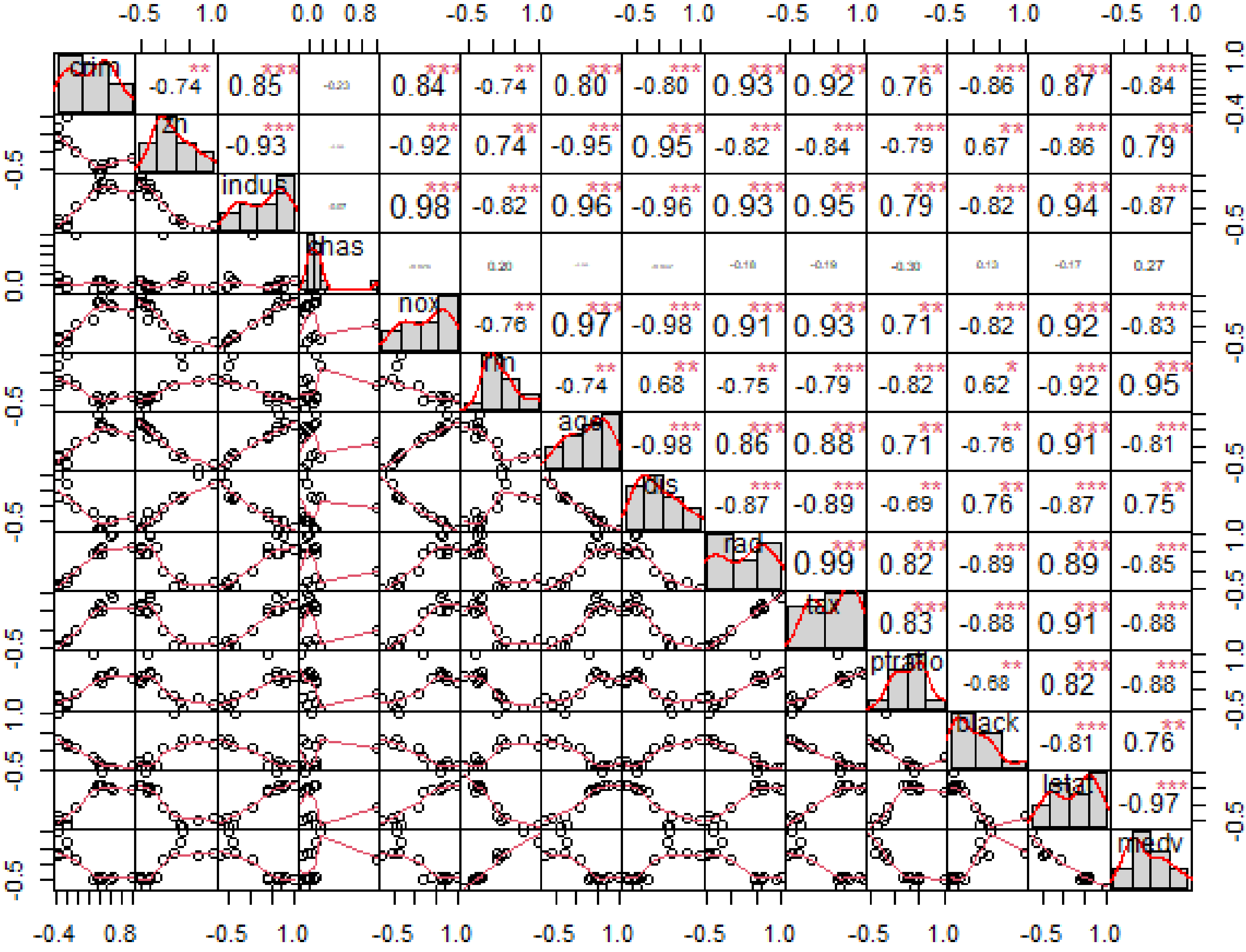
```
> describe(Boston)
        vars   n   mean     sd median trimmed    mad    min    max  range  skew kurtosis   se
crim       1 506   3.61   8.60   0.26    1.68   0.33   0.01  88.98  88.97  5.19    36.60 0.38
zn         2 506  11.36  23.32   0.00    5.08   0.00   0.00 100.00 100.00  2.21     3.95 1.04
indus      3 506  11.14   6.86   9.69   10.93   9.37   0.46  27.74  27.28  0.29    -1.24 0.30
chas       4 506   0.07   0.25   0.00    0.00   0.00   0.00   1.00   1.00  3.39     9.48 0.01
nox        5 506   0.55   0.12   0.54    0.55   0.13   0.38   0.87   0.49  0.72    -0.09 0.01
rm         6 506   6.28   0.70   6.21    6.25   0.51   3.56   8.78   5.22  0.40     1.84 0.03
age        7 506  68.57  28.15  77.50   71.20  28.98   2.90 100.00  97.10 -0.60    -0.98 1.25
dis        8 506   3.80   2.11   3.21    3.54   1.91   1.13  12.13  11.00  1.01     0.46 0.09
rad        9 506   9.55   8.71   5.00    8.73   2.97   1.00  24.00  23.00  1.00    -0.88 0.39
tax       10 506 408.24 168.54 330.00  400.04 108.23 187.00 711.00 524.00  0.67    -1.15 7.49
ptratio   11 506  18.46   2.16  19.05   18.66   1.70  12.60  22.00   9.40 -0.80    -0.30 0.10
black     12 506 356.67  91.29 391.44  383.17   8.09   0.32 396.90 396.58 -2.87     7.10 4.06
lstat     13 506  12.65   7.14  11.36   11.90   7.11   1.73  37.97  36.24  0.90     0.46 0.32
medv      14 506  22.53   9.20  21.20   21.56   5.93   5.00  50.00  45.00  1.10     1.45 0.41
```

Points to be noted:

1. We have 506 entries and 13 feature values (crim, zn, indus, chas, nox, rm, age, dis, rad, tax, ptratio, black, lstat) and 1 target value (medv)
2. There is no Null value or missing value in any feature.
3. Skew value for crim is very high 5.19 and also kurtosis is 36.60. So the data is highly skewed)

Plotting Data

```
library("PerformanceAnalytics")
correlations = cor(Boston)
chart.Correlation(correlations, histogram=TRUE, pch=20,cex = 5)
```

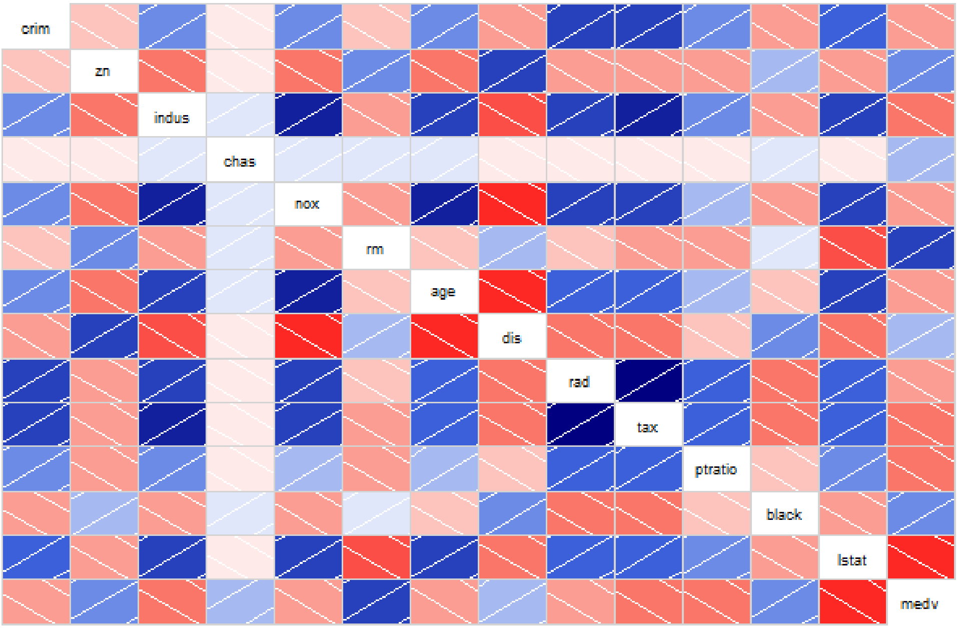


Plot shows distribution of each feature and also correlation between them lower side of matrix is representing graphical and upper side matrix displaying value.

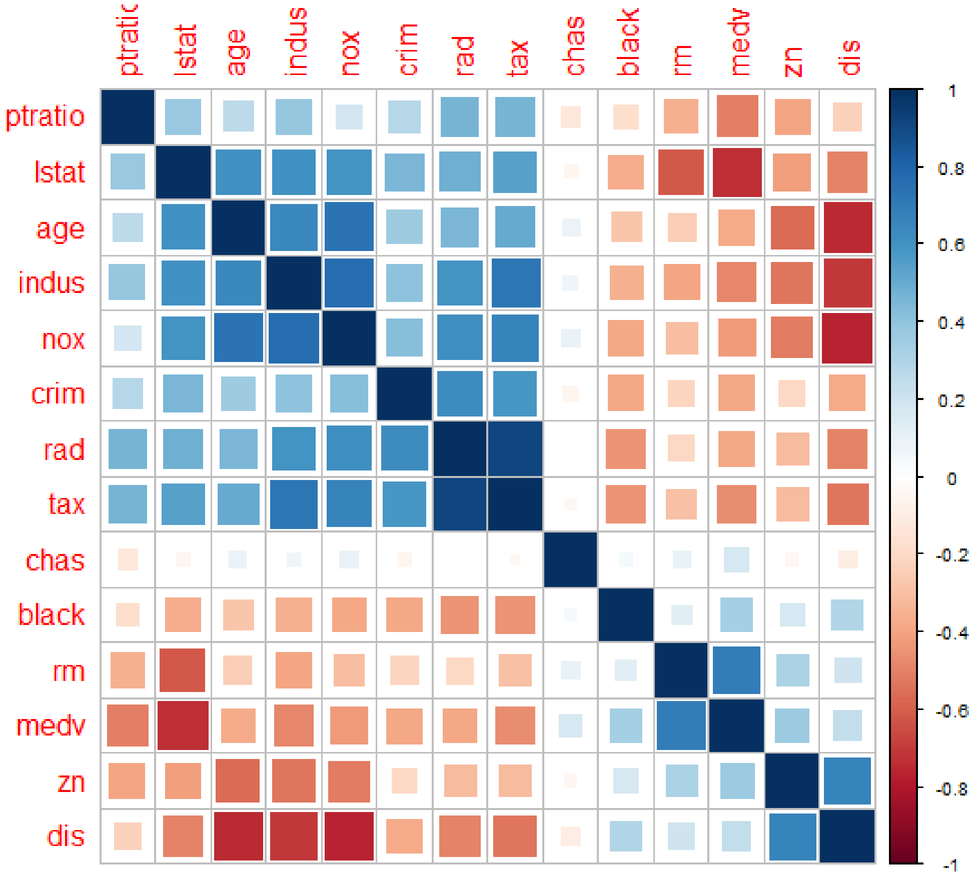
1. When numeric independent variables are highly correlated that can create multi-collinearity problem .

Correlation Matrix

```
library(corrplot)
correlations = cor(Boston)
corrplot(correlations)
corrplot(correlations,order="hclust",method="square",tl.cex = 0.6,cl.cex =0.6)
```

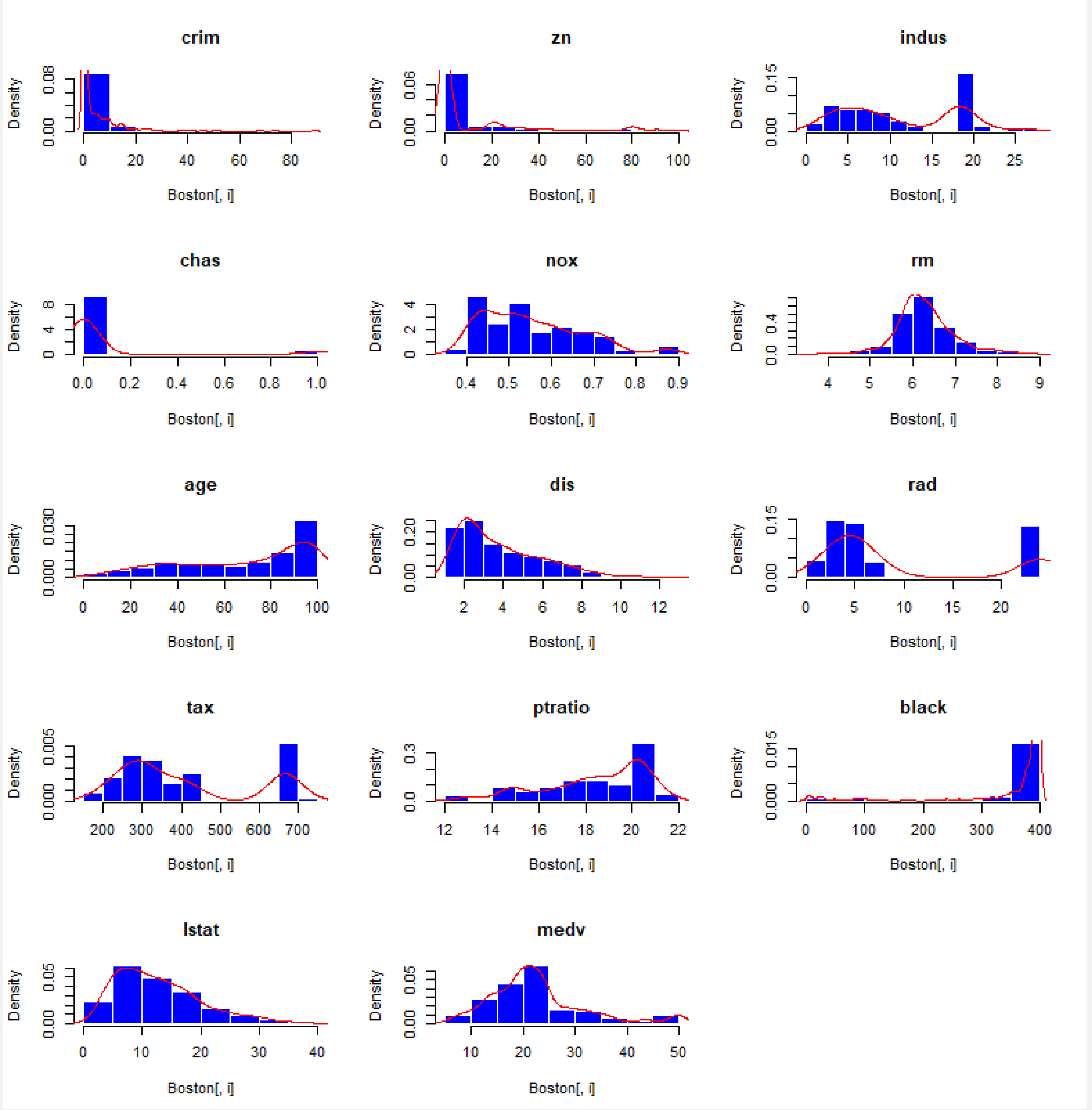


Plot shows Correlation between various features in a form of heat map and dark colours represent high correlation and diagonal lines shows positive or negative correlation.



Distribution of Features

```r
par(mfrow=c(3, 3))
colnames <- dimnames(Boston)[[2]]
for (i in 1:14)
{
  hist(Boston[,i], main=colnames[i], probability=TRUE, col="blue",
       border="white")
  d <- density(Boston[,i])
  lines(d, col="red")
}
```



The only categorical variable chas has more than 80 % 0s and less than 10% 1s, so there are very few housing plots where the land tract bounds the Charles River.

Conclusion

1. We notice that the housing value has a strong positive correlation with rm as expected, as a more spacious house with more rooms would have a higher valuation. medv has a strong negative correlation with lstat meaning an area with lower socioeconomic status naturally has a lower value.
2. We also noticed some very high correlations between the independent variables.
   a. Nox, indus-0.76
   b. Age, nox -0.73

c. Dis, indus - -0.71

d. Dis, nox - -0.77

e. Dis, age - -0.75

f. Tax, indus - 0.72

g. Tax, rad - 0.91

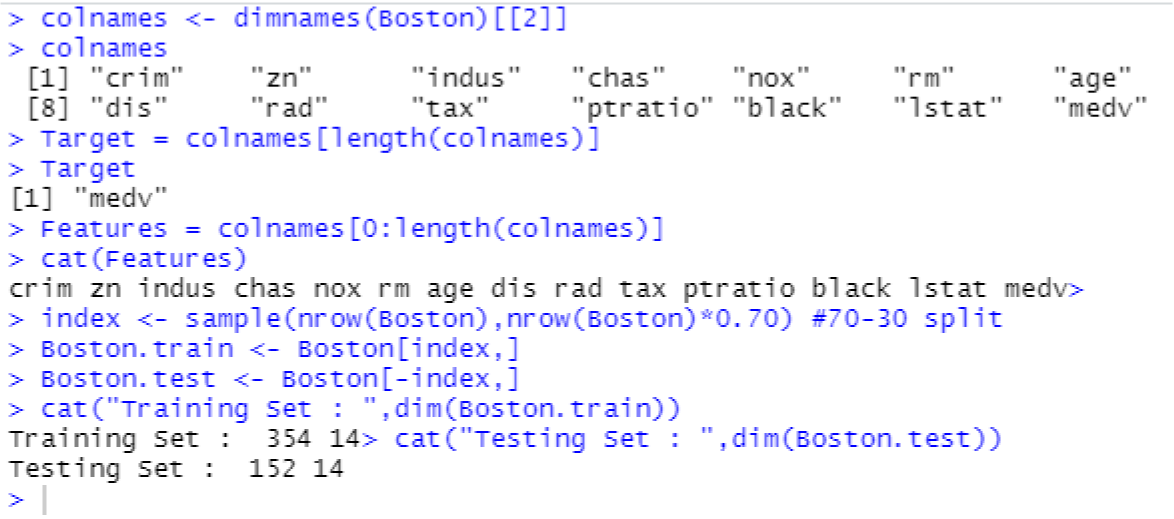
3. The feature with the least correlation to MV is the proximity to Charles River, CHAS.

4. The right skewed distribution suggests that a log transformation would be appropriate. Similarly, the variables crim, dis, nox, zn are found to be right skewed, making log transformations appropriate. The left skewed distribution of ptratio suggests that squaring it could make for a better fit.

Preparing Data:

```
colnames <- dimnames(Boston)[[2]]
colnames
Target = colnames[length(colnames)]
Target
Features = colnames[0:length(colnames)]
cat(Features)

acceptableError = 3
```

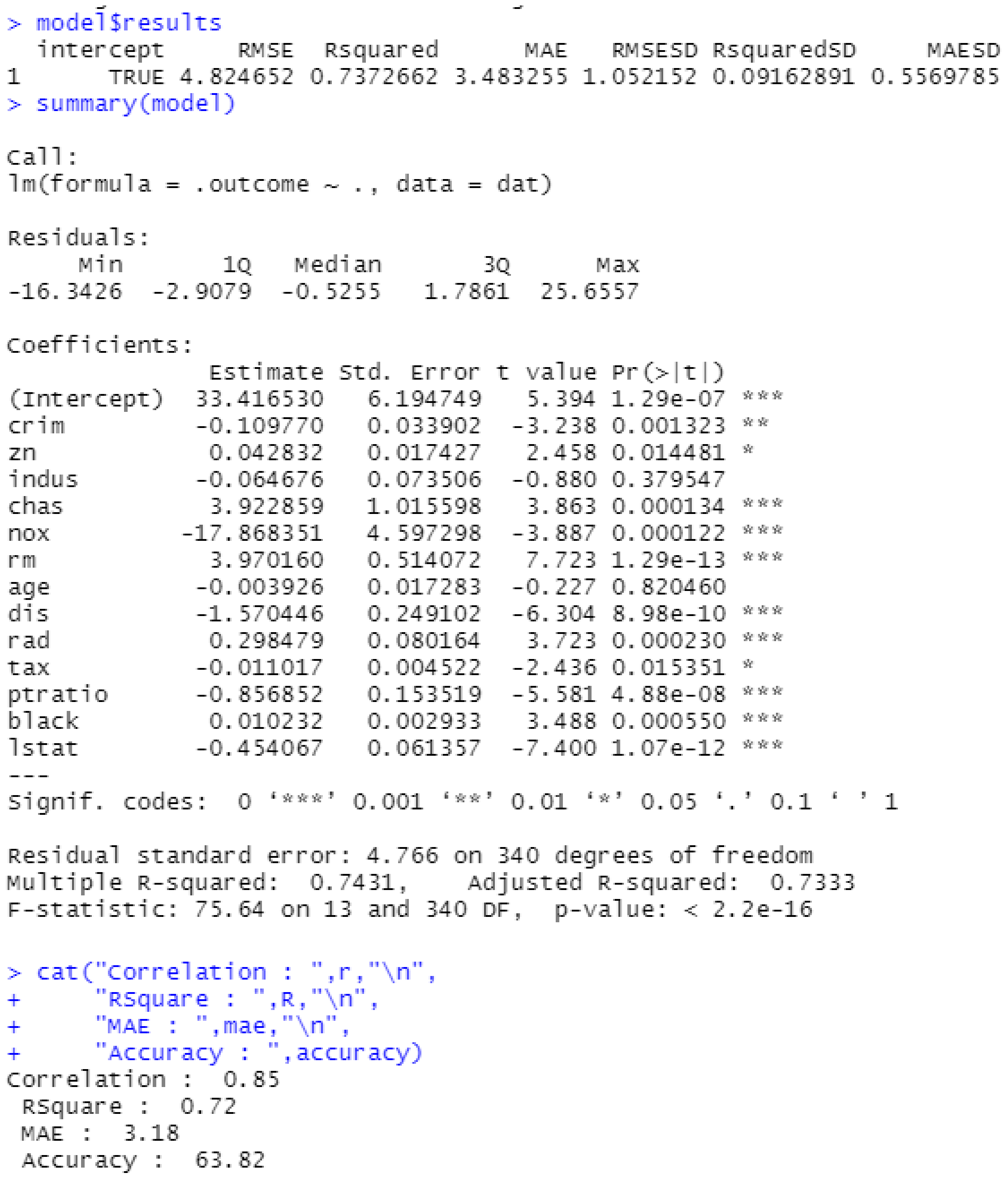
```
index <- sample(nrow(Boston),nrow(Boston)*0.70) #70-30 split
Boston.train <- Boston[index,]
Boston.test <- Boston[-index,]
cat("Training Set : ",dim(Boston.train))
cat("Testing Set : ",dim(Boston.test))
```



```
> colnames <- dimnames(Boston)[[2]]
> colnames
 [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
 [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
> Target = colnames[length(colnames)]
> Target
[1] "medv"
> Features = colnames[0:length(colnames)]
> cat(Features)
crim zn indus chas nox rm age dis rad tax ptratio black lstat medv>
> index <- sample(nrow(Boston),nrow(Boston)*0.70) #70-30 split
> Boston.train <- Boston[index,]
> Boston.test <- Boston[-index,]
> cat("Training Set : ",dim(Boston.train))
Training Set :  354 14> cat("Testing Set : ",dim(Boston.test))
Testing Set :  152 14
> 
```

Generating A linear model

```
#  Cross Validation
library(caret)
CV <- trainControl(method = "repeatedcv",number = 10,repeats = 5,
                      verboseIter = TRUE)
model <- train(medv~.,
               Boston.train,
               method = "lm",
               trControl = CV)

model$results
summary(model)

# Prediction
Actual = as.double(unlist(Boston.test[Target]))
Predicted = predict(model1,Boston.test)

# Model Evaluation

# Correlation
r <- round(cor(Actual,Predicted),2)
# RSquare
R <- round(r * r,2)
# MAE (Mean Absolute)
mae <- round(mean(abs(Actual-Predicted)),2)
# Step 14.4: Accuracy
accuracy <- round(mean(abs(Actual-Predicted) <=acceptableError),4)*100

cat("Correlation : ",r,"\n",
    "RSquare : ",R,"\n",
    "MAE : ",mae,"\n",
    "Accuracy : ",accuracy)
```



```
> model$results
  intercept     RMSE  Rsquared      MAE   RMSESD RsquaredSD     MAESD
1      TRUE 4.824652 0.7372662 3.483255 1.052152 0.09162891 0.5569785
> summary(model)

Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
     Min       1Q   Median       3Q      Max 
-16.3426  -2.9079  -0.5255   1.7861  25.6557 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  33.416530   6.194749   5.394 1.29e-07 ***
crim         -0.109770   0.033902  -3.238 0.001323 ** 
zn            0.042832   0.017427   2.458 0.014481 *  
indus        -0.064676   0.073506  -0.880 0.379547    
chas          3.922859   1.015598   3.863 0.000134 ***
nox         -17.868351   4.597298  -3.887 0.000122 ***
rm            3.970160   0.514072   7.723 1.29e-13 ***
age          -0.003926   0.017283  -0.227 0.820460    
dis          -1.570446   0.249102  -6.304 8.98e-10 ***
rad           0.298479   0.080164   3.723 0.000230 ***
tax          -0.011017   0.004522  -2.436 0.015351 *  
ptratio      -0.856852   0.153519  -5.581 4.88e-08 ***
black         0.010232   0.002933   3.488 0.000550 ***
lstat        -0.454067   0.061357  -7.400 1.07e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.766 on 340 degrees of freedom
Multiple R-squared:  0.7431,	Adjusted R-squared:  0.7333 
F-statistic: 75.64 on 13 and 340 DF,  p-value: < 2.2e-16

> cat("Correlation : ",r,"\n",
+     "RSquare : ",R,"\n",
+     "MAE : ",mae,"\n",
+     "Accuracy : ",accuracy)
Correlation :  0.85 
 RSquare :  0.72 
 MAE :  3.18 
 Accuracy :  63.82
```

1. We can see Accuracy is very low on test data.
2. This model has multi-collinearity.
3. We can sole multi-collinearity by doing regularisation.

With Ridge Regularisation

Using – gnet (general leaner model net)

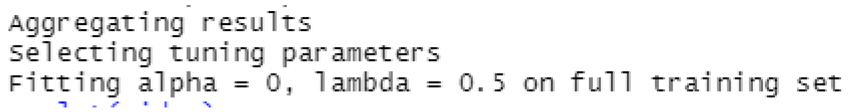
alpha = 0

Lambda = 0.00001 - 1

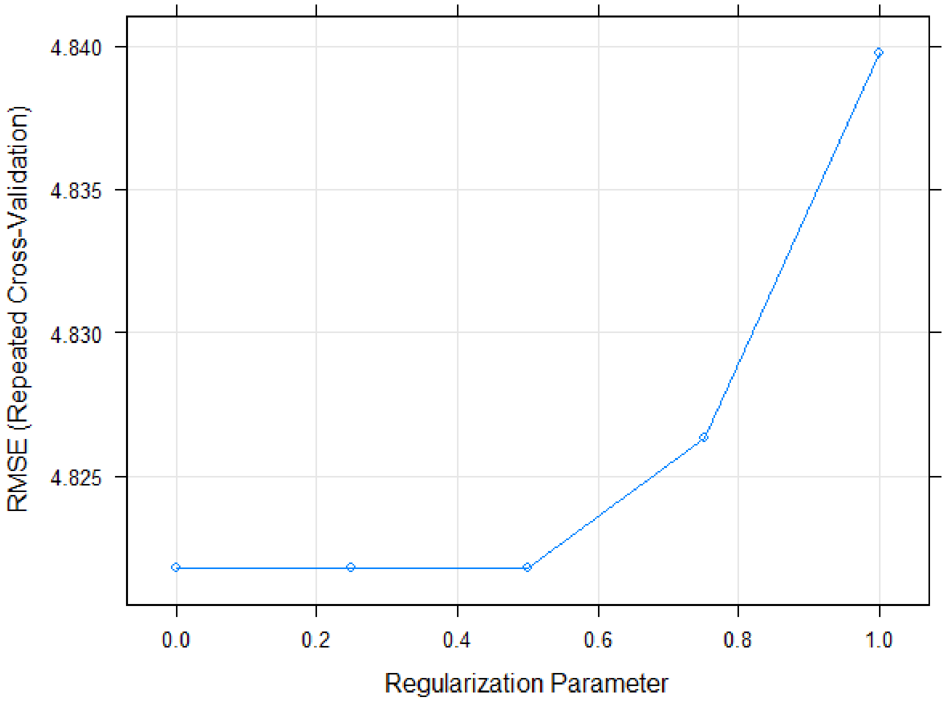
```
ridge <- train(medv~.,
               Boston.train,
               method = 'glmnet',
               tuneGrid = expand.grid(alpha=0,
                                      lambda = seq(0.0001,1,length=5)),
               trControl = CV)
plot(ridge)
plot(varImp(ridge,scale = T))
```

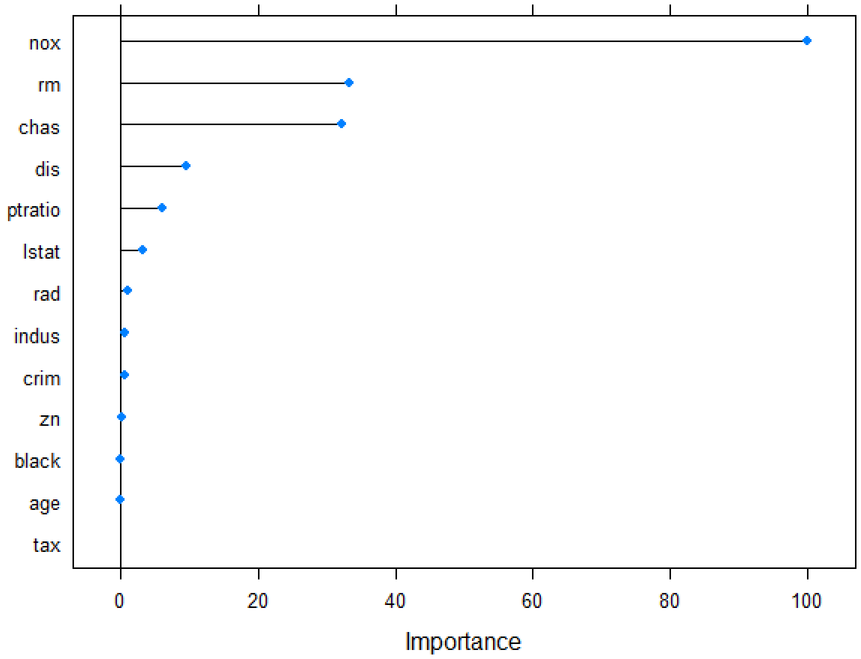
```
Actual = as.double(unlist(Boston.test[Target]))
Predicted = predict(ridge,Boston.test)
```

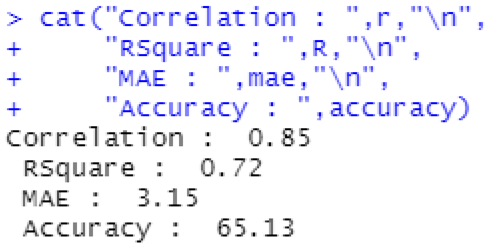
```
# Correlation
r <- round(cor(Actual,Predicted),2)

# RSquare
R <- round(r * r,2)

# MAE (Mean Absolute)
mae <- round(mean(abs(Actual-Predicted)),2)

# Accuracy
accuracy <- round(mean(abs(Actual-Predicted) <=acceptableError),4)*100

cat("Correlation : ",r,"\n",
    "RSquare : ",R,"\n",
    "MAE : ",mae,"\n",
    "Accuracy : ",accuracy)
```



```
Aggregating results
Selecting tuning parameters
Fitting alpha = 0, lambda = 0.5 on full training set
```







```
> cat("Correlation : ",r,"\n",
+     "RSquare : ",R,"\n",
+     "MAE : ",mae,"\n",
+     "Accuracy : ",accuracy)
Correlation :  0.85
 RSquare :  0.72
 MAE :  3.15
 Accuracy :  65.13
```

- Ridge Give improvement in accuracy.

With Lasso Regularisation

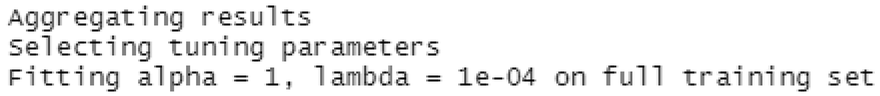
alpha = 1

Lambda = 0.00001 - 1

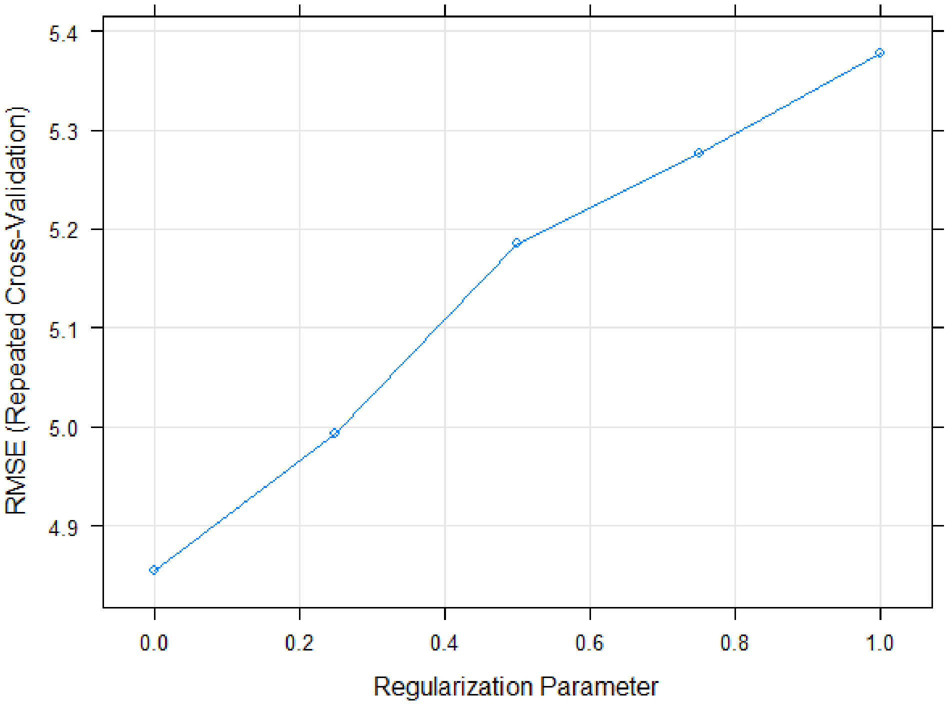
```
lasso <- train(medv~.,
               Boston.train,
               method = 'glmnet',
               tuneGrid = expand.grid(alpha=1,
                                      lambda = seq(0.0001,1,length=5)),
               trControl = CV)
plot(lasso)
plot(varImp(lasso,scale = T))
```

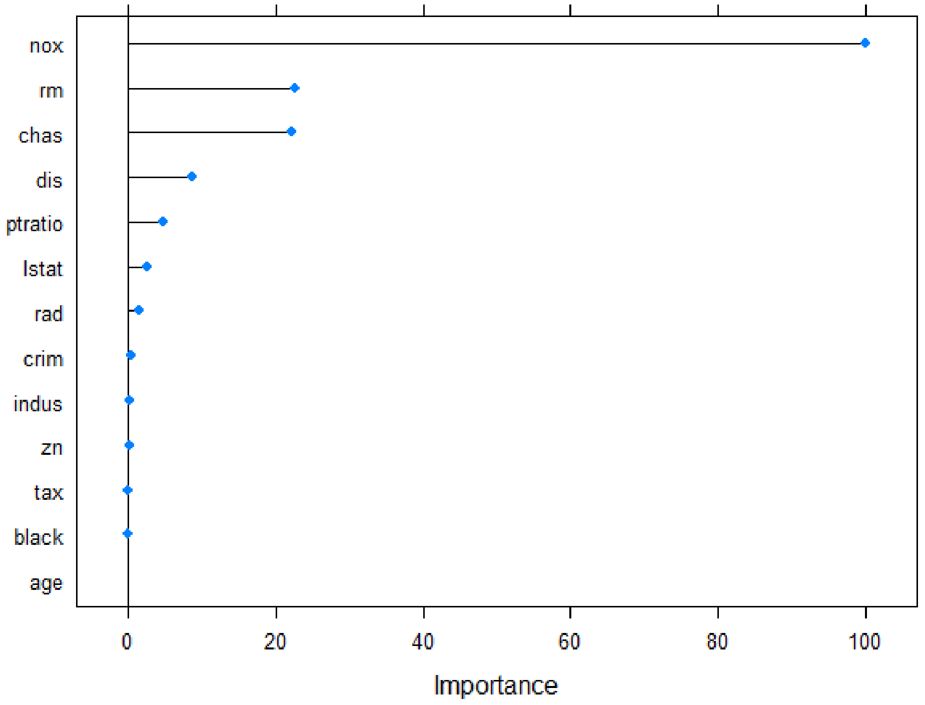
```
Actual = as.double(unlist(Boston.test[Target]))
Predicted = predict(lasso,Boston.test)
```

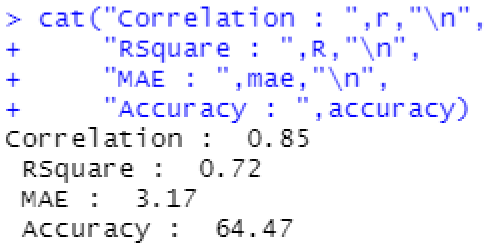
```
# Correlation
r <- round(cor(Actual,Predicted),2)

# RSquare
R <- round(r * r,2)

# MAE (Mean Absolute)
mae <- round(mean(abs(Actual-Predicted)),2)


# Accuracy
accuracy <- round(mean(abs(Actual-Predicted) <=acceptableError),4)*100

cat("Correlation : ",r,"\n",
    "RSquare : ",R,"\n",
    "MAE : ",mae,"\n",
    "Accuracy : ",accuracy)
```



```
Aggregating results
Selecting tuning parameters
Fitting alpha = 1, lambda = 1e-04 on full training set
```







```
> cat("Correlation : ",r,"\n",
+       "RSquare : ",R,"\n",
+       "MAE : ",mae,"\n",
+       "Accuracy : ",accuracy)
Correlation :  0.85
 RSquare :  0.72
 MAE :  3.17
 Accuracy :  64.47
```

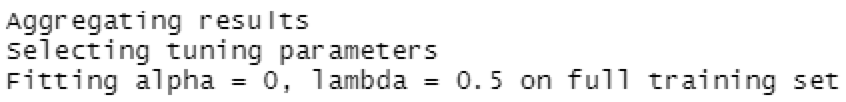
Accuracy of ridge is still better but not much changed.

With Elastic Net Regularisation

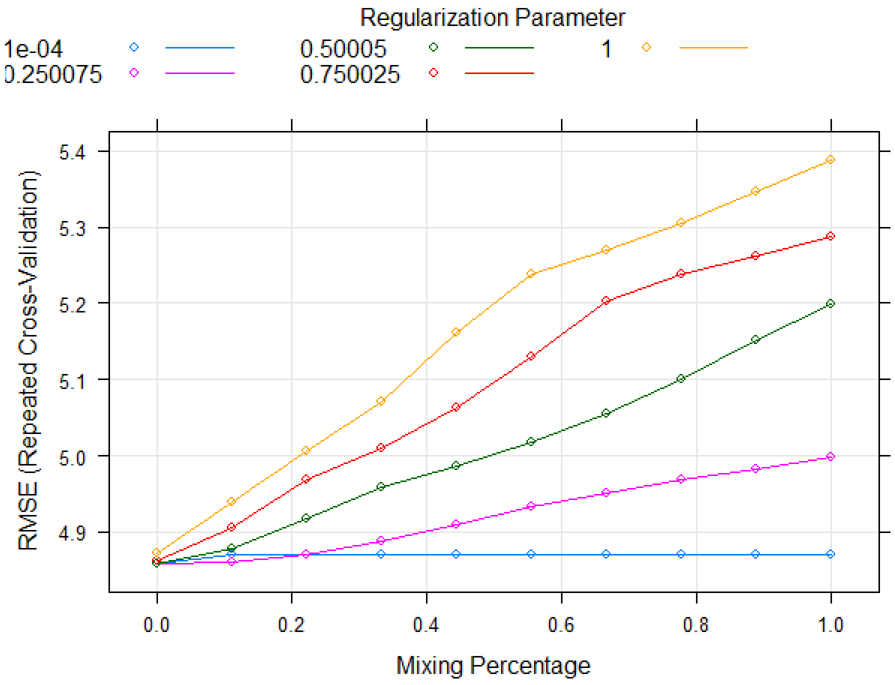
```
EN <- train(medv~.,
               Boston.train,
               method = 'glmnet',
               tuneGrid = expand.grid(alpha=seq(0,1,length=10),
                                      lambda = seq(0.0001,1,length=5)),
               trControl = CV)
plot(EN)
plot(varImp(EN,scale = T))
```

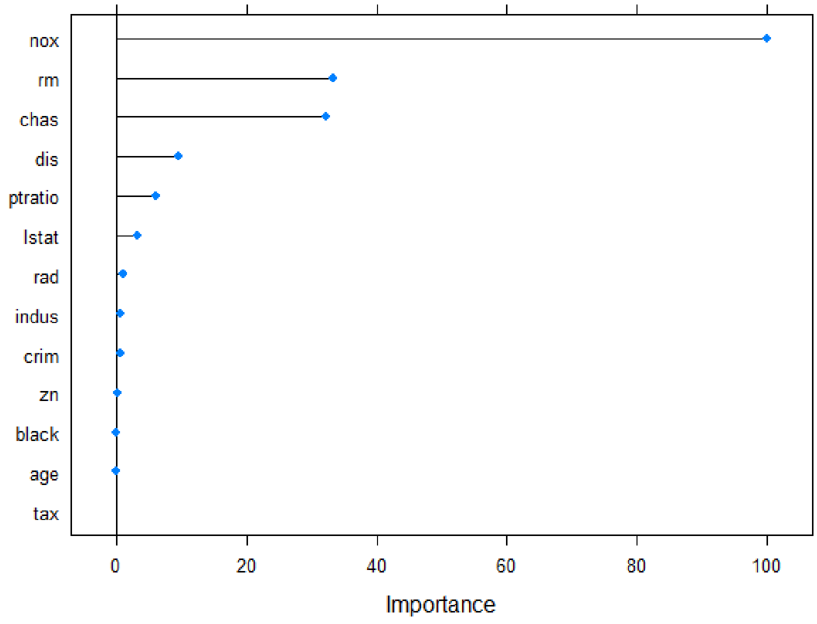
```
Actual = as.double(unlist(Boston.test[Target]))
Predicted = predict(EN,Boston.test)
```

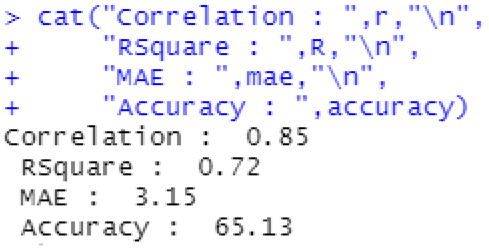
```
r <- round(cor(Actual,Predicted),2)

#  RSquare
R <- round(r * r,2)

# MAE (Mean Absolute)
mae <- round(mean(abs(Actual-Predicted)),2)

# Accuracy
accuracy <- round(mean(abs(Actual-Predicted) <=acceptableError),4)*100

cat("Correlation : ",r,"\n",
    "RSquare : ",R,"\n",
    "MAE : ",mae,"\n",
    "Accuracy : ",accuracy)
```



```
Aggregating results
Selecting tuning parameters
Fitting alpha = 0, lambda = 0.5 on full training set
```





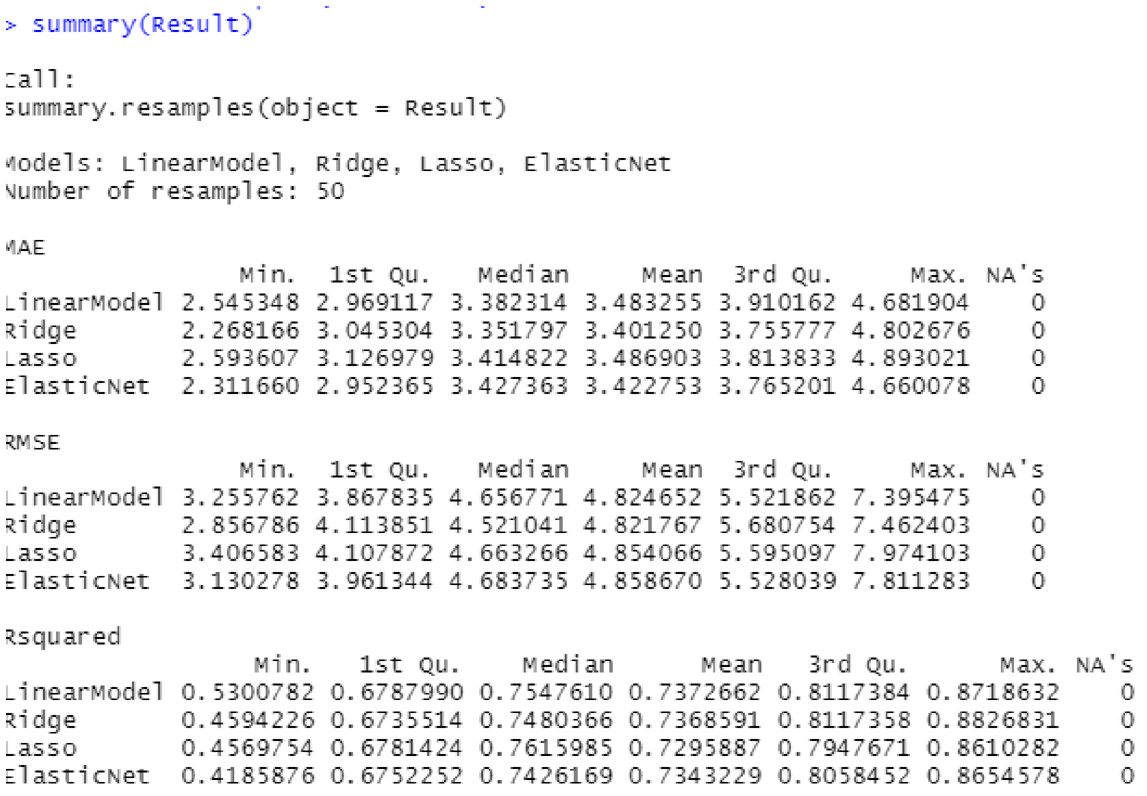


```
> cat("Correlation : ",r,"\n",
+      "RSquare : ",R,"\n",
+      "MAE : ",mae,"\n",
+      "Accuracy : ",accuracy)
Correlation :  0.85 
 RSquare :  0.72 
 MAE :  3.15 
 Accuracy :  65.13
```

- Accuracy for Elastic net is same as Ridge.

Compare All models

```
model_list = list(LinearModel=model,
                  Ridge = ridge,
                  Lasso = lasso,
                  ElasticNet = EN)


Result <- resamples(model_list)

summary(Result)
```



```
> summary(Result)

Call:
summary.resamples(object = Result)

Models: LinearModel, Ridge, Lasso, ElasticNet
Number of resamples: 50

MAE
                Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
LinearModel 2.545348 2.969117 3.382314 3.483255 3.910162 4.681904    0
Ridge       2.268166 3.045304 3.351797 3.401250 3.755777 4.802676    0
Lasso       2.593607 3.126979 3.414822 3.486903 3.813833 4.893021    0
ElasticNet  2.311660 2.952365 3.427363 3.422753 3.765201 4.660078    0

RMSE
                Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
LinearModel 3.255762 3.867835 4.656771 4.824652 5.521862 7.395475    0
Ridge       2.856786 4.113851 4.521041 4.821767 5.680754 7.462403    0
Lasso       3.406583 4.107872 4.663266 4.854066 5.595097 7.974103    0
ElasticNet  3.130278 3.961344 4.683735 4.858670 5.528039 7.811283    0

Rsquared
                 Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
LinearModel 0.5300782 0.6787990 0.7547610 0.7372662 0.8117384 0.8718632    0
Ridge       0.4594226 0.6735514 0.7480366 0.7368591 0.8117358 0.8826831    0
Lasso       0.4569754 0.6781424 0.7615985 0.7295887 0.7947671 0.8610282    0
ElasticNet  0.4185876 0.6752252 0.7426169 0.7343229 0.8058452 0.8654578    0
```

Summary

By Looking at Summary We Can Observe:

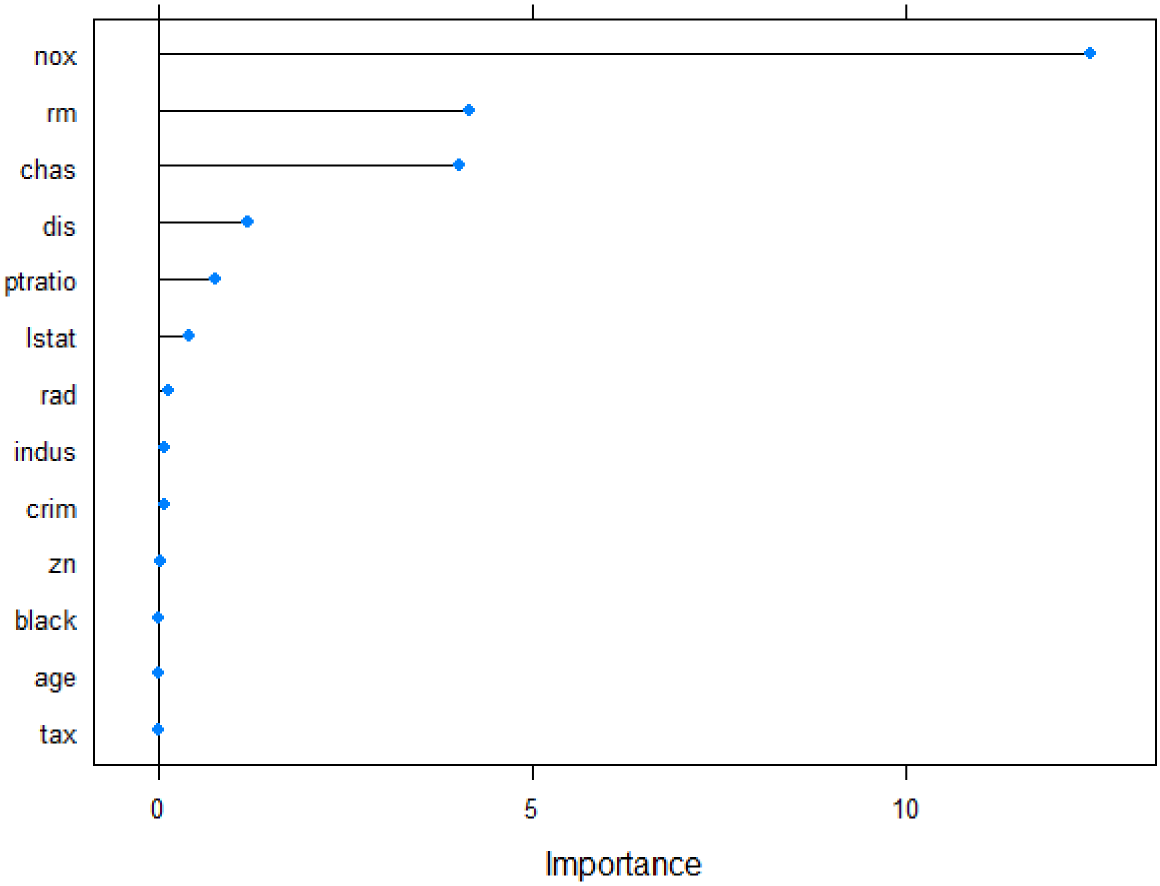
1. Least RMSE mean is 4.821767 for Ridge regression.
2. Maximum R squared value is 0.7368591 for Ridge regression.
3. Also Accuracy score on Test data for ridge and Elastic Net was similar.

Result

- We can Select Ridge regression as our best model with
    - accuracy = 6.15
    - acceptable error = 3
    - mean squared error for train set = 3.15
    - mean squared error for test set = 3.15

So our Feature importance sequence is:

```
plot(varImp(ridge,scale = F))
```


nox
rm
chas
dis
ptratio
lstat
rad
indus
crim
zn
black
age
tax
Importance